

ANEXO EXAMEN DE CERTIFICACIÓN

Plan de Estudio	Desarrollo Aplicaciones Fullstack Java Trainee
Anexo	Caso Classicmodels Sales Force

Caso “Classicmodels Sales Force”

“Classicmodels” es una empresa dedicada a la fabricación y comercialización de modelos a escala de distintos tipos de vehículos motorizados tales como automóviles, motocicletas, trenes y aviones, entre otros. Sus modelos son de alto realismo y de gran calidad, razón por la cual ha logrado posicionarse en el mercado de los coleccionistas a nivel mundial.

La empresa se creó en los años sesentas como una empresa familiar, y debido a la buena calidad de sus modelos, fue rápidamente creciendo. Actualmente, cuenta con oficinas en las principales capitales del mundo, tales como New York, Boston, San Francisco, Paris, Tokyo, Sydeny, y London.

El crecimiento de la empresa no ha estado exento de dificultades, de hecho, el principal reclamo es el atraso en los pedidos debido a que no se lleva un buen control de las existencias y las órdenes. Al respecto, el CEO de la organización ha solicitado al CTO que impulse un proyecto de mejora de los sistemas de inventarios y existencias que ya data de los años 90s y no se le han hecho grandes mejoras.

El CTO está muy entusiasmado con el proyecto y rápidamente ha formado un equipo de proyectos de primera línea del cual usted forma parte como desarrollador full-stack Java. El equipo de proyectos también lo conforma un Jefe de Proyectos, un Diseñador UX/UI, un Diseñador Web, un Analista Funcional, un Desarrollador Mobile, y un Arquitecto de Software.

El proyecto busca, dentro de otras cosas, ordenar el sistema de inventario y de órdenes que se cursan desde las distintas oficinas. A continuación, se listan los requisitos funcionales de alto nivel del sistema:

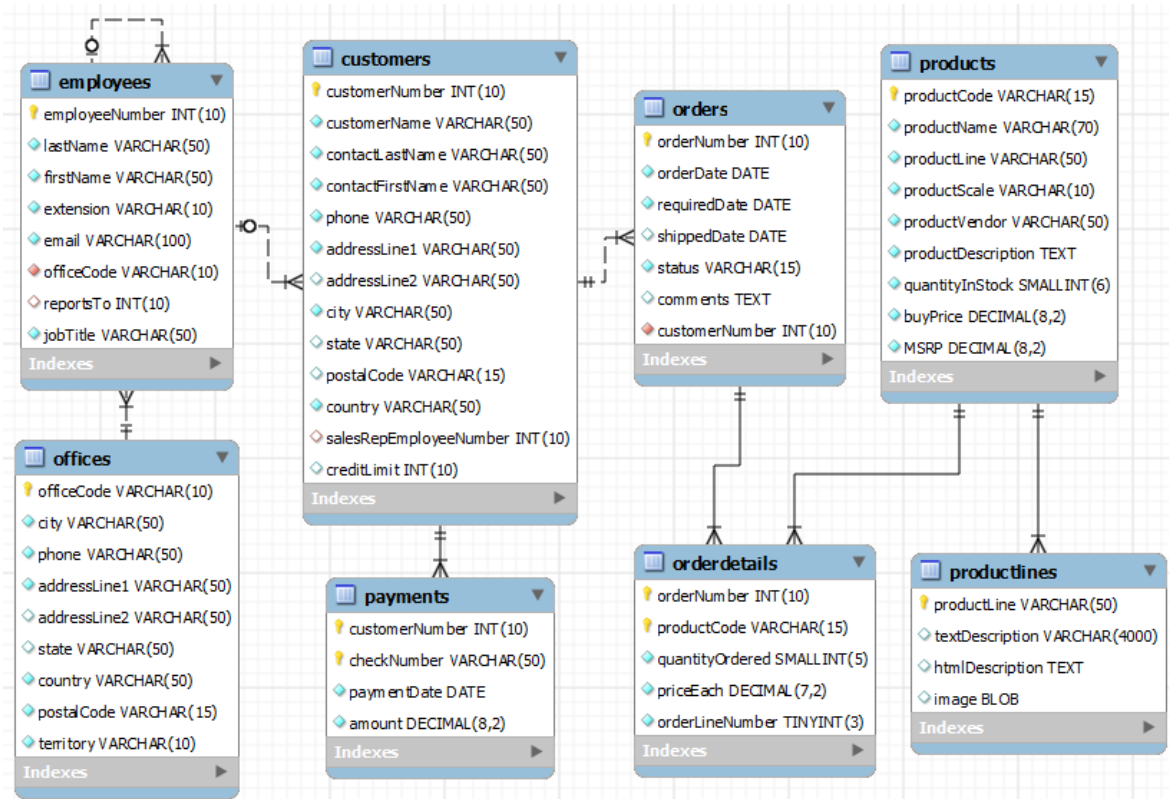
- El sistema debe permitir la consulta de inventarios y existencia de los productos de las distintas líneas (automóviles, motocicletas, trenes, aviones, etc)
- El sistema debe permitir la consulta de las órdenes que se encuentran en proceso
- El sistema debe permitir el cálculo de descuentos y promociones al momento de ingresar una orden

A la fecha, ha transcurrido gran parte del proyecto y se tiene el siguiente avance:

- Ya se cuenta con un prototipo funcional del aplicativo
- Existe un modelo de datos diseñado
- Existe una base de datos con datos de prueba
- Existe una aplicación web desarrollada con Spring Framework que desarrolla algunas de las funcionalidades requeridas

Modelo de Datos

A continuación, se presenta el modelo de datos diseñado por el arquitecto en conjunto con el analista funcional.



Como se puede observar, el modelo contiene las órdenes de la compañía en una relación master-detail entre las tablas *orders* y *orderdetails*.

Por otra parte, existe un maestro de productos (tabla *products*) que mantiene los principales atributos de cada producto, en donde se encuentra el código de producto, nombre, datos de stock (quantityInStock) y precio de costo (buyPrice), entre otros.

Los productos están agrupados en líneas de producto, para así poder administrarlos de mejor manera. La tabla *productlines* entrega la información de cada línea de productos, el cual es utilizado por los category managers para hacer gestión por categoría.

El maestro de oficinas (*offices*) contiene información de cada sucursal. Cada sucursal, a su vez, maneja un grupo de vendedores y managers los cuales se encuentran almacenados en la tabla *employees*. Cada vendedor maneja su propia cartera de clientes, lo cual se ve reflejado en la relación con la tabla *customers*.

La tabla *employees*, no solamente alberga a los vendedores y managers sino toda la estructura de la organización, con todos los miembros de la compañía en una relación jerárquica a través del campo *reportsTo*.

El maestro de clientes (tabla *customers*) contiene la información general de cada uno de los clientes de la compañía y a su vez se relaciona con la tabla de órdenes (*orders*) y de pagos (*payments*).

Requerimientos a Desarrollar

El jefe de proyectos, quien lleva un control meticuloso de las actividades del proyecto, le ha solicitado a Usted que realice las siguientes tareas:

1. Realizar consultas a la base de datos
2. Construir un algoritmo de cálculo de comisiones de venta
3. Construir una unidad de pruebas para verificar los algoritmos de cálculo de comisiones de venta
4. Crear un Monitor de Gestión de Vendedores
5. Crear una API REST que disponibilice la información del monitor de gestión de vendedores

A continuación, se especifica con mayor detalle cada uno de los requerimientos:

1. Realizar consultas a la base de datos

El Chief Sales Manager ha solicitado algunos reportes de información, razón por la cual el jefe del proyecto le ha encargado a usted que realice algunas consultas en la base de datos, para la extracción de cierta información necesaria para el negocio, por mientras se termina el desarrollo de la aplicación. **Para esto, cree un package en su proyecto Java con nombre “consultas”, cree un archivo por cada una de ellas, identificando claramente de qué consulta se trata** (ejm: Consulta-A.sql, Consulta-B.sql, ...etc).

- a) Se requiere una nómina con los datos de los representantes de venta (Sales Rep) junto al país y oficina al cual pertenece. Ordene alfabéticamente por país, ciudad y primer nombre. El reporte debe tener la siguiente forma:

country	city	firstName	lastName	email
Australia	Sydney	Andy	Fixter	afixter@classicmodelcars.com
Australia	Sydney	Peter	Marsh	pmarsh@classicmodelcars.com
Australia	Sydney	Tom	King	tking@classicmodelcars.com
France	Paris	Gerard	Hernandez	ghernande@classicmodelcars.com
France	Paris	Loui	Bondur	lbondur@classicmodelcars.com
France	Paris	Martin	Gerard	mgerard@classicmodelcars.com

- b) Se requiere tener un listado de los clientes que corresponden a empresas de responsabilidad limitada (Ltd), con sus límites de crédito ordenados de mayor a menor. El reporte debe tener la siguiente forma:

customerNumber	customerName	creditLimit ▼ 1
124	Mini Gifts Distributors Ltd.	210500.00
298	Vida Sport, Ltd	141300.00
259	Toms Spezialitäten, Ltd	120400.00
276	Anna's Decorations, Ltd	107800.00
458	Corrida Auto Replicas, Ltd	104600.00
148	Dragon Souvenirs, Ltd.	103800.00

- c) Se requiere tener un listado con los representantes de venta (Sales Rep) y su cartera de clientes (cantidad de empresas que atienden), ordenado de forma descendente desde la cartera más grande. El reporte debe tener la siguiente forma:

employeeNumber	firstName	lastName	count(*) ▼ 1
1401	Pamela	Castillo	10
1504	Barry	Jones	9
1323	George	Vanauf	8
1501	Larry	Bott	8
1370	Gerard	Hernandez	7
1286	Foon Yue	Tseng	7
1337	Loui	Bondur	6
1702	Martin	Gerard	6

- d) Se requiere un ranking de ventas por oficina. Esto es, la cantidad de órdenes que han sido cursadas por los clientes, asociados a los representantes de venta de cada oficina. El listado debe estar ordenado por cantidad de ventas de mayor a menor. El reporte debe tener la siguiente forma:

country	city	count(*) ▾ 1
France	Paris	106
USA	San Francisco	48
UK	London	47
USA	NYC	39

- e) Se requiere un ranking de los clientes que más compraron (monto de la orden, es decir, precio por cantidad pedida de cada producto de la orden) durante el año 2004, ordenado de mayor a menor suma de monto. El reporte debe tener la siguiente manera:

customerName	SumaMontoOrdenes
Euro+ Shopping Channel	340830.87
Mini Gifts Distributors Ltd.	231562.53
Australian Collectors, Co.	127155.96
Land of Toys Inc.	126792.53
Vida Sport, Ltd	108777.92
Handji Gifts& Co	105420.57
AV Stores, Co.	99984.40
Diecast Classics Inc.	98509.25

2. Construir un algoritmo de cálculo de comisiones de venta

Un objetivo importante de este proyecto es desarrollar un nuevo sistema de comisiones para los representantes de venta de forma que incentive una mejor gestión de las ventas. Para esto, se le pide que desarrolle un algoritmo de cálculo de las comisiones que pueda ir extendiéndose a futuro, en la medida que se vayan percibiendo los beneficios de este nuevo sistema de incentivo.

Para lograr lo anterior, se le pide que desarrolle un algoritmo de cálculo de comisiones de venta con distintas lógicas de cálculo. Construya dos algoritmos de cálculo de comisiones, uno que realice **cálculos simples** y otro que realice **cálculos complejos**.

Las reglas del algoritmo de **cálculo simple** son las siguientes:

- Si la suma del monto total de ventas de un representante es superior a U\$ 10.000, entonces la comisión de ventas debe ser del 10%. Caso contrario, debe ser de un 5%

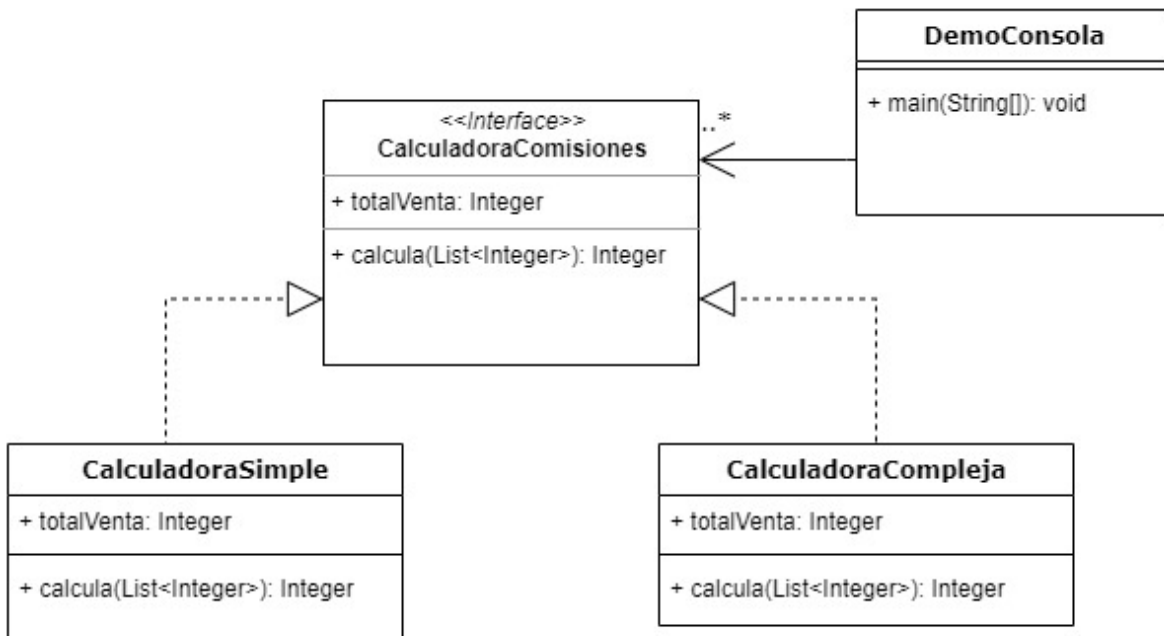
Las reglas del algoritmo de **cálculo complejo** son las siguientes:

- El algoritmo debe aplicar un 5% de comisión fija a todo el monto de venta del representante.

- Si se registra una venta de más de U\$ 5.000 y menos de U\$ 10.000, entonces debe aplicar un 2% adicional de comisión a esa venta
- Si se registra una venta de más de U\$ 10.000, entonces debe aplicar un 3% adicional de comisión de venta
- Si la suma de todas las ventas es superior a 50.000, entonces debe aplicar un 1% adicional a todo el monto de venta.

Ambos algoritmos deben recibir un listado de valores enteros correspondientes al monto de la venta de un vendedor y debe retornar el cálculo del monto de la comisión redondeado (valor entero).

El arquitecto del proyecto le sugiere que realice un diseño de clases similar al del siguiente diagrama:



Para hacer una demostración de los algoritmos, cree una aplicación de consola que genere 5 valores aleatorios entre \$1.000 y \$20.000, que correspondería a la venta de un representante, y que posteriormente realice el cálculo de las comisiones utilizando las dos implementaciones creadas anteriormente (algoritmo simple y complejo). A continuación se presenta un ejemplo de la ejecución:

```

-----
Demostración Calculadora de Comisiones de Venta
-----

```

```

Tomando 5 montos de venta aleatorios...
13543 16523 8172 1341 9832

```

```
Descuento con Algoritmo Simple: 3030
Descuento con Algoritmo Complejo: 3968
```

Considere hacer un diseño de clases polimórfico, mediante interfaces, en donde deberá crear una interfaz y dos clases concretas que implementen dicha interfaz. Una de ellas que implemente el algoritmo de cálculo simple y otra que implemente el algoritmo de cálculo complejo. Asimismo, una clase que sea la que ejecuta la aplicación de consola y genera los montos de venta aleatorios para entregárselo a cada uno de los algoritmos.

Genere dentro de su proyecto en eclipse un package con nombre representativo que tenga las clases mencionadas.

3. Construir una unidad de pruebas para verificar el correcto cálculo de las comisiones

Construya una clase de pruebas en Java que permita verificar el correcto funcionamiento del algoritmo de “cálculo complejo” de comisiones de venta, considerando al menos los siguientes tests:

- Tests que considere casos normales, con distinta cantidad y montos de comisión
- Tests que considere condiciones de borde, por ejemplo, qué pasa cuando viene un valor cero, u otra condición de excepción.

4. Crear monitor para Gestión de Vendedores

Se requiere crear una página web dinámica que permita el listado y búsqueda de vendedores, tal como se detalla en la siguiente imagen mock-up.

Gestion de Vendedores

Oficina

Seleccione Oficina

Sales Manager

Seleccione SM

Buscar

Num Empleado	Nombre	Mail	Oficina	Manager	Acción
1165	Leslie Jennings	ljennings@classicmodelcars.com	Boston	Gerard Bondour	VER
1166	Leslie Thompson	lthompson@classicmodelcars.com	Boston	Gerard Bondour	VER
1167	Julie Firrelli	jfirrelli@classicmodelcars.com	NYC	Gerard Bondour	VER
1168	Steve Patterson	spatterson@classicmodelcars.com	NYC	Gerard Bondour	VER

Se pide:

- a) Desplegar el listado de oficinas en el primer combobox con valores de la base de datos, ordenado alfabéticamente
- b) Desplegar el listado de Sales Managers en el segundo combobox con valores de la base de datos, ordenado alfabéticamente
- c) Desplegar el listado de representantes de venta a partir de la información en la base de datos y de los filtros seleccionados (oficina y salesmanager).

Para realizar el requerimiento, el arquitecto le señala lo siguiente:

- Utilizar jsp y taglibs jstl para el despliegue de la vista (u otra tecnología de vista)
- Utilizar bootstrap para los elementos
- Que los elementos se ajusten a distintos tamaños de pantalla

5. Crear una API REST que disponibilice el listado de representantes de venta

Disponibilice un servicio REST que permita obtener la misma información del monitor de gestión de vendedores. Recuerde que el servicio podría recibir como parámetro la oficina y el sales manager.